

PATTERN- AND MODEL-BASED POWER LINE MONITORING

RELATED APPLICATIONS

This patent application claims priority to and the benefit of the previously filed provisional patent applications "Home Networking," filed on August 17, 1999, and
5 assigned serial no. 60/149,390 [attorney docket no. 1018.050US0], and "Home
Networking System," filed on February 24, 2000, and assigned serial no. 60/184,631
[attorney docket no. 141388.2].

FIELD OF THE INVENTION

This invention relates generally to power line monitoring, and more particularly to
10 an automation system for controlling and monitoring devices and sensors that performs
power line monitoring.

BACKGROUND OF THE INVENTION

Home networking and automation have become more popular. With the number
and complexity of audio/video equipment increasing, some homeowners are interested in
15 operating their equipment more easily. Other homeowners are more concerned about the
security and safety of their homes. These homeowners may want to remotely monitor
their homes, remotely control appliances and other power line devices, and learn when
important events occur. For example, an important event can be the hot water heater
bursting or leaking, or another type of event. Power line devices are devices that connect
20 to the power line, usually through a plug that connects to an electrical outlet.

Currently, there are two popular home networking infrastructures. The first is
phone line networking. To provide in-home networking of computers and computer

peripherals without requiring home rewiring, as is usually required with standard Ethernet networks, the Home Phone line Networking Alliance (HomePNA) was formed to leverage the existing phone lines in homes. More detailed information regarding the HomePNA can be found on the Internet at www.homepna.com. While phone line

5 networking allows homeowners to create small local-area networks (LAN's) within their homes for the purposes of connecting computers and computer peripherals, it has limitations. Significantly, phone line networking typically does not allow homeowners to control appliances, lamps, and other power line devices within their homes.

A second home networking infrastructure is power line networking. Power line

10 networking provides ubiquitous wired connectivity throughout the majority of homes. One type of power line networking is known as X10. X10 is a communications protocol that allows for remotely controlling power line devices, such as lamps and appliances. More detailed information regarding the X10 protocol can be found on the Internet at <ftp://ftp.scruz.net/users/cichlid/public/x10faq>.

15 Current power line networking, such as X10 networking, is limited. The X10 protocol, for example, provides only a rudimentary and low-level framework for controlling and monitoring power line devices. The framework generally does not allow for sophisticated and complex device control applications. While automation systems employing existing X10 technology can be implemented using computers, more typically

20 the systems are implemented with relatively less intelligent control centers that only govern a limited number of power line devices, in a limited manner. When computers are used, the resulting systems are still far from ideal. They may be difficult to use, and may not be reliable or robust against equipment failures and crashes.

An intelligent, reliable, and robust automation system that overcomes these
Ser. No. 09/641,489
problems is described in the cofiled patent application entitled "Automation System for
Controlling and Monitoring Devices and Sensors" [attorney docket no. 1018-050US1].
The system can include power line monitoring, to determine whether power line devices
5 are malfunctioning, or whether malicious intrusions are being attempted on the system.
Such power line monitoring is the subject of this patent application.

SUMMARY OF THE INVENTION

The invention relates to power line monitoring. The monitoring can be performed
in conjunction with an automation system designed to control and monitor devices and
10 sensors. The devices can include power line devices, such as lamps, appliances,
audio/video equipment, and other devices that connect to the power line of a house or
other building. The sensors can include sensors for detecting the occurrence of
emergency-related and other events. For example, a water sensor located near a hot
water heater can detect whether the heater has burst, or is leaking.

15 The power line monitoring can be model-based, pattern-based, or both. Model-
based power line monitoring uses a model of acceptable power line activity. Activity that
does not conform to the model is tagged as indicating a potential problem. Pattern-based
power line monitoring uses patterns of unacceptable power line activity. Activity that
matches one of the patterns is also tagged as indicating a potential problem.

20 The invention can be used in conjunction with an automation system, such as the
automation system described in this patent application. The invention can also be used in
conjunction with another type of system, or independently from a system. The phrase
"power line monitoring system" encompasses any type of system in which power line

monitoring is performed, such as an automation system. The invention encompasses automation systems, automation system architectures, power line monitoring systems, and methods of varying scopes. Other aspects, embodiments and advantages of the invention, beyond those described here, will become apparent by reading the detailed description and by referencing the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a pictorial diagram of an example house in which an automation system can be implemented.

FIG. 2 is a topological diagram of the automation system of FIG. 1.

FIG. 3 is a diagram of a software architecture that can implement the automation system of FIGs. 1 and 2.

FIG. 4 is a diagram showing how devices, sensors, and objects are registered with the look-up services of the software architecture of FIG. 3.

FIG. 5 is a diagram showing how an object of FIG. 4 can be addressed in two different ways.

FIG. 6 is a diagram showing the soft-state store of FIG. 3 in more detail.

FIG. 7 is a diagram showing an abstract, high-level view of the software architecture of FIG. 3.

FIG. 8 is a diagram showing a power line monitoring daemon as one example of the system management daemons of FIG. 3, according to an embodiment of the invention.

FIG. 9 is a diagram showing how the power line monitoring daemon of FIG. 8 detects patterns from the power line, according to an embodiment of the invention.

FIG. 10 is a flowchart of a method showing how one embodiment monitors the power line.

FIG. 11 is a diagram of an example computerized device that can be used to implement an automation system.

5

DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description of exemplary embodiments of the invention, reference is made to the accompanying drawings that form a part hereof, and in which is shown by way of illustration specific exemplary embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. Other embodiments may be utilized, and logical, mechanical, electrical, and other changes may be made without departing from the spirit or scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

15 Automation System Hardware Architecture

FIG. 1 shows a pictorial diagram 100 of an example house 102 in which an automation system can be implemented. The house 102 includes a garage 104, a kitchen 106, a family room 108, a master bedroom 110, and a den 114, among other rooms. Connections coming into and distributed throughout the house 102 include a phone line 116, a power line 118, and an Internet connection 120. The phone line 116 allows residents of the house 102 to place and receive phone calls. The power line 118 provides electrical power to the house 102. The Internet connection 120 connects the house 102 to

the Internet. The Internet connection 120 can be a fast, always-on broadband connection, such as a cable modem connection or a Digital Subscriber Loop (DSL) line. The Internet connection 120 may also be a slow, dial-up connection that utilizes the phone line 116.

Not specifically called out in FIG. 1 is that a network has been set up within the house 102. The network is of the type that connects computers and computer peripherals with one another. For example, the network may be an Ethernet network, using dedicated wiring, or using the existing phone line 116. The network may also be a wireless Ethernet network, or another type of network. For purposes of descriptive clarity, this network is referred to as the backbone network in the detailed description.

The backbone network is distinct from a power line network, for example.

Each of the rooms of the house 102 has a number of network adapters and electrical outlets, as well as other types of components. The garage 104 has four components pertinent to the automation system. The video camera 122 allows remote monitoring of whether the garage door 124 is open or closed. The camera 122 is preferably directly connected to the backbone network set up within the house 102. The network adapter 128, as well as other network adapters throughout the house 102, provide for connectivity to the backbone network. The garage door opener 130 controls the opening and closing of the garage door 124, and is connected to the backbone network through the network adapter 128.

Electrical power devices that may be controlled using the automation system can be plugged into the electrical outlet 126, or into other electrical outlets throughout the house 102. Electrical power devices are also referred to as power line devices. Power line devices include appliances, lamps, audio/video equipment, and other types of devices

The den 114 is the room of the house 102 in which the heart of the automation system is located. There are four computing devices 174, 176, 178, and 180. The computing devices may be desktop or laptop computers, for example. The computing device 174 serves as the gateway device, through which the backbone network set up in the house 102 is connected to the Internet connection 120. The devices 176, 178, and 180 provide the hardware on which the software architecture of the automation system is implemented in a distributed manner. The software architecture is described in detail in a subsequent section of the detailed description. There is more than one such device for redundancy and reliability purposes. The devices 174, 176, 178, and 180 connect to the backbone network through the network adapter 182, and can receive power through the electrical outlet 184.

The four computing devices 174, 176, 178, and 180 can be located throughout the house 102, instead of in a single room, such as the den 114. This may be desirable where one or more of these computing devices also serves as a UAP. Furthermore, if a circuit breaker for the room where one of the computing devices is located trips, only one computing device is affected. The automation system will still be able to operate over the other, unaffected computing devices.

The RF bridge 144 of the den 114 allows the RF devices 142 and 148, in the kitchen 106 and the family room 108, respectively, to communicate with other devices on the backbone network set up in the house 102. The RF bridge 144 is connected to the backbone network through the network adapter 146, and receives power through the electrical outlet 186. There are two other bridges in the den 114, the IR bridge 188, and the power bridge 192. The infrared (IR) bridge 188 allows the IR device 190, and other

09641556 "081700
007F80 955F950

IR devices, to communicate with devices on the backbone network. The IR device 190 sends IR signals to and receives IR signals from the IR bridge 188, and vice-versa.

Examples of the IR device 190 include a video-cassette recorder (VCR) and a remote control, although the device 190 can be any type of device. IR signals differ from RF

5 signals in that they require a direct line of sight between the sender and the receiver, unlike RF signals. The IR bridge 188 receives power from the electrical outlet 194, and is connected to the backbone network through the network adapter 196.

The power bridge 192 allows devices connected to the power line 118 of the house 102 via electrical outlets to communicate with devices on the backbone network.

10 The power bridge 192 is connected to the backbone network through the network adapter 182, and through the electrical outlet 198 receives power and communicates with the devices connected to the power line 118. For example, the lamp 168 in the master bedroom 110 can be controlled and monitored by the automation system. The device adapter 166 situated between the lamp 168 and the electrical outlet 164 sends and
15 receives signals over the power line 118. The power bridge 192 transfers these signals from the power line 118 to the backbone network set up in the house 102.

While the automation system has been shown in FIG. 1 as implemented in a house, the system can also be implemented in other types of buildings as well. For example, the automation system can be implemented in an office building, a church, a
20 store, a mall, or another type of building. The automation system can also be implemented without regard to a physical structure, such as a building. The components controlled and used by the automation system of FIG. 1 are representative components,

and are not all required to implement the automation system. As an example, the IR device 190 and the RF devices 142 and 148 may be omitted.

FIG. 2 shows a diagrammatic topology of the automation system of FIG. 1, providing another view of the system. The automation system is called out as the system 200 in FIG. 2. The backbone network 202 is preferably an Ethernet network, implemented over a dedicated line or over the phone line 116 of FIG. 1. The system devices 204 include the devices 174, 176, and 178. The devices 204 connect to the backbone network 202 through the network adapter 182. The device 174 is the gateway device that connects to the Internet connection 120. The user access points (UAP's) 206 include the UAP's 101, 103, and 105. The UAP's 206 are preferably directly connected to the backbone network 202. Likewise, the thermostat 160 is directly connected to the backbone network 202, whereas the water sensor 138 is connected to the backbone network 202 through the network adapter 140. The audio/video (A/V) devices 156 are connected to the A/V bridge 158, which is connected to the backbone network 202 through the network adapter 154. The A/V bridge 158 enables the A/V devices 156 to communicate with devices on the backbone network 202.

The power bridge 192 is connected to the backbone network 202 through the network adapter 182. Two instances of the same network adapter 182 are shown in FIG. 2 for illustrative clarity. The power bridge 192 is connected to the power line 118 through the electrical outlet 198. Smart power devices 208 directly connect to the power line 118 through corresponding electrical outlets 210, and can directly communicate with the power bridge 192. By comparison, non-intelligent power devices require interstitial device adapters between them and their corresponding electrical outlets. For example,

the lamp 168 requires the device adapter 166 between it and the electrical outlet 164 for the automation system to control and monitor the lamp 168. Moving to the right of the power bridge 192 on the backbone network 202 in FIG. 2, the garage door opener 130 is connected to the backbone network 202 through the network adapter 128. The video camera 122 is directly connected to the backbone network 202.

The infrared (IR) bridge 188 is connected to the backbone network 202 through the network adapter 196, while the radio frequency (RF) bridge 144 is connected to the backbone network 202 through the network adapter 146. The IR bridge 188 enables the IR devices 212, such as the IR device 192, to communicate with devices on the backbone network 202. Likewise, the RF bridge 144 enables the RF devices 214, such as the RF devices 142 and 148, to communicate with devices on the backbone network 202.

Automation System Software Architecture

FIG. 3 is a diagram 300 showing a software architecture 302 for the automation system described in the previous section of the detailed description. The software architecture 302 specifically has three layers, a system infrastructure layer 304, an application layer 306, and a user interface layer 308. The software architecture 302 is preferably implemented over the system devices 204 of FIG. 2, such as the devices 176, 178, and 180. An overview of each layer of the architecture 302 is described in turn. The architecture 302, which is the central and critical aspect of the software architecture, is then described in more detail.

The system infrastructure layer 304 includes look-up services 310, a publication/subscription eventing component 312, system management daemons 314, and a soft-state store 316. The soft-state store 316 manages the lifetime and replication of

00641556-084700

soft-state variables. The publication/subscription eventing component 312 enables objects, daemons, programs, and other software components to subscribe to events related to changes in the soft-state store 316. The look-up services 310 interact with devices and sensors of the automation system, which are indicated by the arrow 318.

- 5 Specifically, the look-up services 310 include a name-based look-up service (NBLS) 320, and an attribute-based look-up service (ABLS) 322. The ABLS 322 maintains a database of available devices, and supports queries based on device attributes. The device attributes can include device type and physical location, among other attributes. The NBLS 320 maintains a database of running instances of objects, and supports name-to-
- 10 object address mapping. The system management daemons 314 of the system infrastructure layer 304 detect failures of devices, and initiate recovery actions.

- The application layer 306 includes automation applications 324, device objects 326, and device daemons 328. There are two types of automation applications 324, device-control applications, and sensing applications. Device-control applications
- 15 receive user requests as input, consult the look-up services 310 to identify the devices and the device objects 326 that should be involved, and perform actions on them to satisfy the requests. The device objects 326 correspond to the devices and sensors identified by the arrow 318. The device objects 326 encapsulate device- and network-specific details of their corresponding devices, and present interfaces for them, such as method calls.
- 20 Examples of the device objects 326 include camera objects for taking snapshots and recording video clips, and garage door opener objects for operating garage doors.

Sensing applications monitor environmental factors, and take actions when a monitored event occurs. The sensing applications subscribe to events through the

eventing component 312. Device daemons 328 interact with the devices and sensors identified by the arrow 318, and independently act as proxies for them. For example, a device daemon for a sensor can monitor sensor signals, and update appropriate soft-state variables in the soft-state store 316 to trigger events.

- 5 The user interface layer 308 provides user access to the system infrastructure layer 304 and the application layer 306. The user interface layer 308 has three parts, a web browser interface 330, a voice-recognition interface 332, and a text-based natural language parser interface 334. The browser interface 330 enables the user to browse through available devices, select devices based on attributes, and control the devices.
- 10 The text-based natural language parser interface 334 is based on a vocabulary appropriate to an automation system, while the voice-recognition interface 332 employs voice recognition technology based on the same vocabulary.

- The user interface layer 308 preferably supports remote automation. For example, when the Internet connection 120 is an always-on connection, the browser
- 15 interface 330 can be used to access the automation system from remote locations. The natural language parser interface 334 provides an email-based remote automation interface. The email daemon 336 periodically retrieves email through the Internet connection 120, and parses automation-related requests contained in the email. The daemon 336 passes the requests to the automation applications 324, and optionally sends
- 20 reply email confirming that the requested actions have taken place. Known digital signature and data encryption technologies can be used to ensure the security of the email. If the user has a mobile phone 338 that supports text messaging, the email daemon 336 can alert the user with text messages when predetermined events occur. The voice-

announced to the ABLS 322 when they are plugged in and switched on, through the device daemons 328 that act as proxies for the devices 404. Device objects 326 for the dynamic devices 404 are instantiated when an application requests to control the devices 404. The objects 326 can persist for a length of time, so that repeated requests do not
5 require repeated instantiation. The device objects 326 are instantiated by the NBLS 320. Smart devices 208 perform their own registration with the ABLS 322. Computation objects 406 are instantiated by the NBLS 320, and require a software component or service referred to as the computation object installer 420 to register with the ABLS 322.

FIG. 5 is a diagram 500 showing how one embodiment addresses an object 502 in two different ways. The object 502 can be one of the device objects 326 of FIG. 4, or one of the computation objects 406 of FIG. 4. The object 502 has one or more synchronous addresses 304, and one or more asynchronous addresses 306. The synchronous addresses 504 can include an address in the form of a marshaled distributed-object interface pointer, or another type of reference that enables real-time communication with the object 502.
15 The asynchronous addresses 306 can be in the form of a queue name, a marshaled handle to a queue, or other address. The asynchronous addresses 306 are used to asynchronously communicate with the object 502 when it is temporarily unavailable or too busy, or when synchronous communication is otherwise not desired.

Referring back to FIG. 4, in summary, the ABLS 322 of the look-up services 310
20 maintains a database of available devices and sensors. The ABLS 322 supports queries based on combinations of attributes, and returns unique names of the devices and sensors that match the queries. By allowing identification of devices and sensors by their attributes and physical locations, instead of by their unique addresses, the ABLS 322

Sub
Q1
10

enables user-friendly device naming. For example, the user can identify a device as “the lamp on the garage side of the kitchen,” instead of by its unique address. The NBLS 320 of the look-up services 310 maps unique names to object instances identified by those names. The NBLS 320 is optionally extensible to allow mapping of a unique name to multiple object instances. Both the ABLS 322 and the NBLS 320 are robust against object failures and non-graceful termination because they are based on the soft-state store 316 of FIG. 3.

FIG. 6 is a diagram 600 showing one embodiment of the soft-state store (SSS) 316 in more detail. The SSS 316 uses a persistent, or non-volatile, store 608, and a volatile store 606. The persistent store 608 is used in addition to the volatile store 606 to save soft-state variables over failures, such as system crashes. The persistent store 608 can be a hard disk drive, non-volatile memory such as flash memory, or other non-volatile storage. The volatile store 606 can be volatile memory, such as random-access memory, or other volatile storage.

The SSS 316 ultimately receives heartbeats 602, and stores them as soft-state variables in either the persistent store 608 or the volatile store 606. The heartbeats 602 are periodic refreshes from devices, sensors, objects, and daemons, so that the automation system knows they are still operating. The heartbeats 602 include device heartbeats 610, sensor heartbeats 612, object heartbeats 614, and daemon heartbeats 616. The refresh rates of the heartbeats 602 vary by their type. The daemon heartbeats 616 may be received over intervals of seconds. The object heartbeats 614 may be received over intervals of tens of seconds to minutes. The sensor heartbeats 612 may be received over

intervals of minutes to hours. The device heartbeats 610 may be received over intervals
from hours to days.

5 *Sub 02* The device heartbeats 610 and the sensor heartbeats 610 are received by the SSS 316 through the ABLs 322, while the object heartbeats 614 are received by the SSS 316 through the NBLs 320. The SSS 316 directly receives the daemon heartbeats 616. When an entity does not send a heartbeat as required by its refresh rate, the entity ultimately times out and is removed from the ABLs 322 and the NBLs 320. An entity in this context refers to a device, sensor, object, or daemon.

10 The SSS 316 preferably performs soft-state variable checkpointing. For a given refresh rate threshold, heartbeats that occur above the threshold, and thus are updated with high frequency, remain in the volatile store 606, to decrease overhead. Recovery of these high-frequency heartbeats from failure of the SSS 316 is through new refreshes. Conversely, heartbeats that occur below the threshold, and thus are updated with low frequency, are persisted in the persistent store 608. Recovery of these low-frequency
15 heartbeats from failure of the SSS 316 is through restoration of the persistent soft-state variables in the store 608. This is because waiting for the next heartbeat may take too long. Downtime of the SSS 316 is preferably treated as missing refreshes for the soft-state variables.

20 The publication/subscription eventing component 312 allows subscriptions to events resulting from the change, addition, or deletion of the soft-state variables maintained by the SSS 316. The subscribers can include applications, daemons, and other software components of the automation system. The eventing component 312 sends events to subscribers when the kinds of changes to the SSS 316 match their

corresponding event subscriptions. The component 312 receives the changes in the soft-state variables from the SSS 316. From these changes, it formulates events as necessary.

As a summary of the software architecture described in this section of the detailed description, FIG. 7 is a diagram 1000 showing a high-level view of the architecture. The software architecture 302 resides between the power line 118 and the Internet connection 120. The architecture 302 abstracts the manner by which the devices connected to the power line 118 are accessed and controlled. The Internet connection 120 provides for remote, off-site access and control of devices connected to the power line 118, through the architecture 302. The notification path 1008 indicates that notifications from the devices connected to the power line 118 move up to the software architecture 302. The notifications can also move up to the Internet connection 120 in the case of remote, off-site access and control of the devices. The control path 1010 indicates that control of the devices originates either from the Internet connection 120 or at the architecture 302, and moves down to the power line 118 to reach the devices.

Power Line Monitoring

FIG. 8 is a diagram 700 showing one embodiment of the system management daemons 314 of FIG. 3 in more detail. In particular, the system management daemons include a power line monitoring daemon 702. The power line monitoring daemon 702 detects reliability, security, and other problems with automation system devices that use the power line. The daemon 702 can use pattern-based detection for detecting unacceptable power line activity, model-based detection for detecting acceptable power line activity, or both. Pattern-based detection employs a database of unacceptable power line patterns that, if detected, trigger an event. For example, faulty devices and external

interferences may produce meaningless repetitions or interleaving of commands. By comparison, model-based detection employs a model of acceptable power line patterns. Power line patterns that do not conform to the model also trigger an event. The daemon 702 is referred to generically as a mechanism.

5 FIG. 9 is a diagram 800 showing how one embodiment uses the power line monitoring daemon 702 to detect problems on the power line 118. The daemon 702 monitors the power line 118 for problems that result from intrusions 802, atypical behaviors 804, and interferences 806. The daemon 702 matches patterns from the power line 118 against unacceptable power line patterns stored in the pattern database 808. The
10 daemon 702 also tests the patterns against the pattern model 810 of acceptable power line patterns. If matching the patterns against the unacceptable power line patterns stored in the database 808 yields a positive match, or testing the patterns against the model 810 of acceptable power line patterns yields a negative test, the daemon 702 generates an event. The event corresponds to the situation that an unacceptable pattern has been detected on
15 the power line 118. Other daemons, objects, and programs can subscribe to the event through the eventing component 312, which is not specifically called out in FIG. 9.

 The monitoring daemon 702 maintains a log file 812 of all detected power line patterns. The patterns stored in the log file 812 include both acceptable and unacceptable power line patterns. An analysis tool 814 can be used by a user to determine whether to
20 add new unacceptable power line patterns to the database 808, based on the patterns stored in the log file 812. The analysis tool 814 can also be used to determine whether the model 810 of acceptable power line patterns should be modified, based on the patterns stored in the log file 812.

bad pattern, which consists of K identical function commands transmitted consecutively. They also capture the case where more than K identical function commands are transmitted consecutively. As another example, a second pattern reduces to detecting the following expression on the power line within ΔT time.

$$5 \quad (\Sigma^*; ((\exists y (Address_Command_Unitcode=y)^M) + (\exists z (Function_Command=z)^N)))^L; \Sigma^* \quad (2)$$

The expression $Address_Command_Unitcode=y$ is an expression that matches events that are address commands with unit code y . $(\exists y \dots)$ denotes the scope of parameter y .

10 Upon detecting the first pattern (1), the power line monitoring daemon 702 queries the ABLS 322 of FIGs. 3, 4, and 6 for the X10 receiver gating the transceiver for house code x . The daemon 702 tries to power it off to see if the bad pattern stops. Upon detecting the second pattern (2), the daemon 702 tries to locate and power off all power line bridges one by one to see if the problem is caused by a faulty bridge. The power line
15 bridges, for example, can include the bridge 192 of FIGs. 1 and 2. In either case, if the problem is not eliminated, the daemon 702 suspects an intruder has caused the pattern.

For model-based detection of acceptable power line activity, power line transmissions are monitored to detect whether they satisfy a predetermined model of good behavior. The predetermined model is initially constructed by training a model
20 based on known acceptable power line activity. For example, the model can be a probabilistic model, a statistical model, a Bayesian network, a Support Vector Machine (SVM), a classifier, or another type of model. Once it has been trained, the model can be tested against collected data, and modified as necessary to produce the optimal monitoring results.

The model can be continually retrained within the automation system on an ad hoc basis, or by a vendor on a more widespread basis, so that it becomes more accurate over time. For example, with respect to the former, power line monitoring that detects unknown, and presumably bad, behavior can be alerted to a user, who can make a
5 decision as to whether the behavior is unacceptable. If it is acceptable, then the model can recalibrate itself so that this type of behavior will not be triggered as unacceptable when detected in the future. As another example, with respect to the latter, as the vendor obtains more power line activity data, it can retrain the model, and issue periodic updates.

Pattern-based detection of unacceptable power line activity and model-based
10 detection of acceptable power line activity can be performed independent of each other, and independent of an automation system. For example, pattern-based detection of unacceptable power line activity and model-based detection of acceptable power line activity can be performed independent of the automation system described in previous sections of the detailed description.

15 FIG. 10 is a flowchart of a method 1100 showing how one embodiment monitors power line activity. The method 1100 can be performed by the power line monitoring daemon 702 of FIGs. 8 and 9. In 1102, a power line pattern of activity is detected. The power line pattern is tested in 1104 against a database of known unacceptable power line patterns, against a model of acceptable power line patterns, or against both. The database
20 can be the database 808 of FIG. 9, while the model can be the model 810 of FIG. 9. If the detected pattern is not found in the database, or does satisfy the model, then the method 1100 proceeds from 1106 to 1108, where the detected power line pattern is deemed acceptable. However, if the detected pattern is found in the database, or does not

004730" 9554960

satisfy the model, the method 1100 proceeds from 1106 to 1110, corresponding to the situation where the pattern has been deemed unacceptable. An appropriate predetermined action is performed in 1110, or an appropriate event is generated. For example, a user can be alerted if the power line monitoring is being performed in conjunction with an automation system.

Example Computerized Device

The automation system can be implemented within a computerized environment having one or more computerized devices. The diagram of FIG. 11 shows an example computerized device 1400. The device 1400 can implement one or more of the system devices 204 of FIG. 2, or one or more of the user access points 206 of FIG. 2. The example computerized device 1400 can be, for example, a desktop computer, a laptop computer, or a personal digital assistant (PDA). The automation system may be practiced with other computer system configurations as well, including multiprocessor systems, microprocessor-based or programmable consumer electronics, network computers, minicomputers, and mainframe computers. The system may be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network.

The device 1400 includes one or more of the following components: processor(s) 1402, memory 1404, storage 1406, a communications component 1408, input device(s) 1410, a display 1412, and output device(s) 1414. For a particular instantiation of the device 1400, one or more of these components may not be present. For example, a PDA may not have any output device(s) 1414. The description of the device 1400 is to be used

as an overview of the types of components that typically reside within such a device, and is not meant as a limiting or exhaustive description.

The processor(s) 1402 may include a single central-processing unit (CPU), or a plurality of processing units, commonly referred to as a parallel processing environment.

5 The memory 1404 may include read-only memory (ROM) and/or random-access memory (RAM). The storage 1406 may be any type of storage, such as fixed-media storage devices and removable-media storage devices. Examples of the former include hard disk drives, and flash or other non-volatile memory. Examples of the latter include tape drives, optical drives like CD-ROM drives, and floppy disk drives. The storage devices
10 and their associated computer-readable media provide non-volatile storage of computer-readable instructions, data structures, program modules, and other data. Any type of computer-readable media that can store data and that is accessible by a computer can be used.

The device 1400 may operate in a network environment. Examples of networks
15 include the Internet, intranets, extranets, local-area networks (LAN's), and wide-area networks (WAN's). The device 1400 may include a communications component 1408, which can be present in or attached to the device 1400. The component 1408 may be one or more of a network card, an Ethernet card, an analog modem, a cable modem, a digital subscriber loop (DSL) modem, and an Integrated Services Digital Network (ISDN)
20 adapter. The input device(s) 1410 are the mechanisms by which a user provides input to the device 1400. Such device(s) 1410 can include keyboards, pointing devices, microphones, joysticks, game pads, and scanners. The display 1412 is how the device 1400 typically shows output to the user. The display 1412 can include cathode-ray tube

